

Claims

1 1. A method for sharing resources on a multithreaded CPU
2 capable of executing a plurality of threads, the method comprising:
3 deferring a yield of a first thread executing on the multithreaded
4 CPU while waiting for at least a second thread executing on the multithreaded
5 CPU to become ready to yield;
6 yielding the first thread in response to at least the second thread
7 becoming ready to yield.

1 2. The method according to claim 1, further comprising
2 monitoring the plurality of threads for an occurrence.

1 3. The method according to claim 2, wherein the occurrence is
2 a spin lock or an idle loop.

1 4. The method according to claim 2, further comprising
2 making a yield call in response to the occurrence.

1 5. The method according to claim 1, further comprising
2 marking storage of the first thread in response to receiving the yield call to
3 indicate that the first thread is ready to yield.

1 6. The method according to claim 1, further comprising
2 spinning the first thread while waiting for at least the second thread to become
3 ready to yield.

1 7. The method according to claim 1, further comprising
2 abandoning the yield call in response to detecting an event.

1 8. The method according to claim 7, wherein the event is a
2 time-out or an external interrupt.

1 9. The method according to claim 7, further comprising
2 returning control of the first thread to an operating system in response to
3 detecting the event.

1 10. The method according to claim 9, further comprising
2 saving the state of the operating system in response to detecting that at least
3 the second thread is ready to yield.

1 11. The method according to claim 1, further comprising idling
2 at least the first and second threads within a common virtual space in response
3 to at least the second thread being ready to yield.

1 12. The method according to claim 11, further comprising
2 idling all threads executing on the multithreaded CPU within the common
3 virtual space.

IBM ROC920010252US1

1 13. A method for yielding a thread within a multithreaded
2 CPU data processing system, wherein each of a plurality of threads executing
3 on a multithreaded CPU must execute within a common virtual space, the
4 method comprising:

5 deferring a yield of a thread while at least a subset of the
6 plurality of threads yield;

7 abandoning the yield of the thread in response to detecting an
8 event while the yield is deferred.

1 14. The method according to claim 13, further comprising
2 yielding the thread after the subset of threads yield, if the subset of threads
3 yield prior to the event.

1 15. The method according to claim 13, wherein the event is
2 selected from among a group consisting of: a time-out, an I/O interrupt and a
3 combination thereof.

1 16. An apparatus comprising:
2 a computer having a multithreaded CPU, wherein the CPU is
3 configured to execute a plurality of threads; and
4 a program resident in the computer, the program configured to defer a yield of
5 a first thread of the plurality while waiting for at least a second thread of the
6 plurality to become ready to yield; and further to initiate the yield of the first
7 thread in response to at least the second thread of the plurality becoming ready
8 to yield.

1 17. The apparatus according to claim 16, wherein the program
2 initiates monitoring the plurality of threads for an occurrence.

1 18. The method according to claim 17, wherein the occurrence
2 is a spin lock or an idle loop.

1 19. The apparatus according to claim 17, wherein the program
2 initiates a yield call in response to the occurrence.

1 20. The apparatus according to claim 16, wherein the program
2 initiates marking storage of the first thread in response to receiving the yield
3 call to indicate that the first thread is ready to yield.

1 21. The apparatus according to claim 16, wherein the program
2 initiates spinning the first thread while waiting for at least the second thread of
3 the plurality to become ready to yield.

1 22. The apparatus according to claim 16, wherein the program
2 initiates abandoning the yield call in response to detecting an event.

1 23. The apparatus according to claim 22, wherein the event is a
2 time-out or an external interrupt.

1 24. The apparatus according to claim 22, wherein the program
2 initiates returning control of the first thread to an operating system in response
3 to detecting the event.

1 25. The apparatus according to claim 24, wherein the program
2 initiates saving the state of the operating system in response to detecting that at
3 least the second thread is ready to yield.

1 26. The apparatus according to claim 16, wherein the program
2 initiates idling at least the first and second threads of the plurality within a
3 common virtual space in response to at least the second thread of the plurality
4 being ready to yield.

1 27. The apparatus according to claim 26, wherein the program
2 initiates idling all threads of the plurality of threads within the common virtual
3 space.

1 28. A method for yielding a thread within a multithreaded
2 CPU data processing system, wherein each of a plurality of threads of a
3 multithreaded CPU must execute within a common virtual space, the method
4 comprising:

5 deferring a yield of a thread while at least a subset of the
6 plurality of threads yield;
7 abandoning the yield of the thread in response to detecting an
8 event while the yield is deferred.

1 29. The method according to claim 28, further comprising
2 yielding the thread after the subset of threads yield, if the subset of threads
3 yield prior to the event.

1 30. The method according to claim 28, wherein the event is
2 selected from among a group consisting of: a time-out, an I/O interrupt and a
3 combination thereof.

- 1 31. A program product, comprising:
- 2 (a) a program for yielding a thread within a
- 3 multithreaded CPU data processing system, wherein each of a
- 4 plurality of threads that execute on a multithreaded CPU must
- 5 execute within a common virtual space, wherein the program is
- 6 configured to defer a yield of a first thread of the plurality while
- 7 waiting for at least a second thread of the plurality to become
- 8 ready to yield; and further to initiate the yield of the first thread
- 9 in response to at least the second thread becoming ready to
- 10 yield; and
- 11 (b) a signal bearing medium bearing the first program.
- 1 32. The program product of claim 31, wherein the signal
- 2 bearing medium includes at least one of a recordable medium and a
- 3 transmission-type medium.